Subject: [PATCH] virtualization of sysv msg queues is incomplete
Posted by Kirill Korotaev on Mon, 08 Oct 2007 11:04:04 GMT
View Forum Message <> Reply to Message

Virtualization of sysv msg queues is incomplete:
msg_hdrs and msg_bytes variables visible from userspace are global.
Let's make them per-namespace.

Signed-Off-By: Alexey Kuznetsov <alexey@openvz.org>
Signed-Off-By: Kirill Korotaev <dev@openvz.org>

---
 include/linux/ipc.h | 2 ++
 ipc/msg.c | 21 ++++++++++++-----------
 2 files changed, 12 insertions(+), 11 deletions(-)

```
--- ./include/linux/ipc.h.ve1012 2007-10-08 14:35:40.000000000 +0400
+++ ./include/linux/ipc.h 2007-10-08 14:40:31.000000000 +0400
@@ -111,6 +111,8 @@ struct ipc_namespace {
  int  msg_ctlmax;
  int  msg_ctlmnb;
  int  msg_ctlmni;
+ atomic_t msg_bytes;
+ atomic_t msg_hdrs;

  size_t  shm_ctlmax;
  size_t  shm_ctlall;
--- ./ipc/msg.c.ve1012 2007-10-08 14:35:40.000000000 +0400
+++ ./ipc/msg.c 2007-10-08 14:41:41.000000000 +0400
@@ -66,9 +66,6 @@ struct msg_sender {
 #define SEARCH_NOTEQUAL  3
 #define SEARCH_LESSEQUAL 4

-static atomic_t msg_bytes = ATOMIC_INIT(0);
-static atomic_t msg_hdrs = ATOMIC_INIT(0);
-
 static struct ipc_ids init_msg_ids;

 #define msg_ids(ns) (*((ns)->ids[IPC_MSG_IDS]))
@@ -89,6 +86,8 @@ static void __msg_init_ns(struct ipc_nam
  ns->msg_ctlmax = MSGMAX;
  ns->msg_ctlmnb = MSGMNB;
  ns->msg_ctlmni = MSGMNI;
+ atomic_set(&ns->msg_bytes, 0);
+ atomic_set(&ns->msg_hdrs, 0);
  ipc_init_ids(ids);
 }
```

```
@@ -277,10 +276,10 @@ static void freeque(struct ipc_namespace
   struct msg_msg *msg = list_entry(tmp, struct msg_msg, m_list);

   tmp = tmp->next;
-  atomic_dec(&msg_hdrs);
+  atomic_dec(&ns->msg_hdrs);
   free_msg(msg);
  }
- atomic_sub(msq->q_cbytes, &msg_bytes);
+ atomic_sub(msq->q_cbytes, &ns->msg_bytes);
  security_msg_queue_free(msq);
  ipc_rcu_putref(msq);
 }
@@ -447,8 +446,8 @@ asmlinkage long sys_msgctl(int msqid, in
   mutex_lock(&msg_ids(ns).mutex);
   if (cmd == MSG_INFO) {
    msginfo.msgpool = msg_ids(ns).in_use;
-   msginfo.msgmap = atomic_read(&msg_hdrs);
-   msginfo.msgtql = atomic_read(&msg_bytes);
+   msginfo.msgmap = atomic_read(&ns->msg_hdrs);
+   msginfo.msgtql = atomic_read(&ns->msg_bytes);
   } else {
    msginfo.msgmap = MSGMAP;
    msginfo.msgpool = MSGPOOL;
@@ -719,8 +718,8 @@ long do_msgsnd(int msqid, long mtype, vo
   list_add_tail(&msg->m_list, &msq->q_messages);
   msq->q_cbytes += msgsz;
   msq->q_qnum++;
-  atomic_add(msgsz, &msg_bytes);
-  atomic_inc(&msg_hdrs);
+  atomic_add(msgsz, &ns->msg_bytes);
+  atomic_inc(&ns->msg_hdrs);
  }

  err = 0;
@@ -824,8 +823,8 @@ long do_msgrcv(int msqid, long *pmtype,
    msq->q_rtime = get_seconds();
    msq->q_lrpid = task_tgid_vnr(current);
    msq->q_cbytes -= msg->m_ts;
-   atomic_sub(msg->m_ts, &msg_bytes);
-   atomic_dec(&msg_hdrs);
+   atomic_sub(msg->m_ts, &ns->msg_bytes);
+   atomic_dec(&ns->msg_hdrs);
    ss_wakeup(&msq->q_senders, 0);
    msg_unlock(msq);
    break;
```

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers